Defense Technical Information Center Compilation Part Notice

ADP012027

TITLE: N-sided Surface Generation from Arbitrary Boundary Edges

DISTRIBUTION: Approved for public release, distribution unlimited

This paper is part of the following report:

TITLE: International Conference on Curves and Surfaces [4th], Saint-Malo, France, 1-7 July 1999. Proceedings, Volume 1. Curve and Surface Design

To order the complete compilation report, use: ADA399461

The component part is provided here to allow users access to individually authored sections of proceedings, annals, symposia, etc. However, the component should be considered within the context of the overall compilation report and not as a stand-alone technical report.

The following component part numbers comprise the compilation report:
ADP012010 thru ADP012054

N-sided Surface Generation from Arbitrary Boundary Edges

Kiyotaka Kato

Abstract. This paper discusses a general theory and an implementation method for generating a surface patch with concave edges, holes, ridges and valleys in CAD/CAM applications. The surface generation method, which has been proposed to create an N-sided patch with holes, is first reviewed. Such surfaces are generally classified as transfinite surfaces, in which a surface is interpolated to span given curves. In the proposed method, each boundary edge defined in a 2-dimensional domain has an appropriate blending function. The function is defined so that the derivatives are 0 on the edges, and the function values are 1 on one edge and 0 on the other edges, and each edge in the 3-dimensional space is blended smoothly. A revised method is also introduced in this paper. The previously proposed method has some problems in that a surface may not be generated appropriately for concave edges, and the surface has to be manipulated manually if it has holes. This causes distortion and overlap in mapping from a 2D domain into 3D space. In the new method, the blending function is revised, and the boundary edges in the 2D domain are obtained from the edges in the 3D space beforehand. Thus, it is shown that an N-sided patch with concave edges, holes, ridges and valleys can be suitably generated.

§1. Overview

It seemed that the study of surface generation was almost complete after the development of the NURBS (Non-Uniform Rational B-Spline) surface, and many commercial CAD system used the NURBS surface as a unified surface in their systems. However, it is now being recognized that the NURBS surface has some limitations, and is not suitable for some actual cases. In one such case, there is a problem with the generation of an n-sided surface patch. It is rather hard to generate a surface patch for arbitrary topology with the NURBS surface. Besides the development of 4-sided patches, n-sided surfaces have also been studied. The methods developed can be classified into three classes: the recursive subdivision method using polyhedrons, the multiple patch method in which a surface is represented by plural 4-sided patches, and the single surface

Curve and Surface Design: Saint-Malo 1999

Pierre-Jean Laurent, Paul Sablonnière, and Larry L. Schumaker (eds.), pp. 173–182.

ISBN 0-8265-1356-5.

patch method in which a surface patch is represented as just one patch [1]. This paper refers to the single patch method.

A single patch method has been proposed which generates a surface patch from an arbitrary shape and a number of edges together with holes [2]. It was suggested that some relations are needed between the shape of the boundary edges and the shape of the 2D-definition boundary. The method of surface generation is a generalized one, but it was found that a surface cannot be generated well in some cases. An illegal surface is generated when the boundary has a concave shape and the surface has a hole. Distortion or overlap is caused in mapping from a 2D definition space into the 3D space.

Sabin calls the method of surface generation from boundary edges a "transfinite surface" in contrast with the one which is characterized by a finite number of control points. He argued a general theory, and proposed a two sided surface patch and a surface with holes [3-5]. The two sided surface interpolates two given Bezier curves in a 2D definition space so that it forms a smooth surface without singularity. He also tried to resolve this problem from 3D into 2D by using a dynamic model with some constraints to generate a surface patch with holes.

For this same purpose, this paper proposes a method of surface generation which is flexibile in generating a surface from such boundary edges so as not to cause twists and overlaps. The second section of this paper reviews the theory about pre-proposed surface generation. In the third section, the problems of the conventional method are discussed. After that, a new method of resolving these problems is described. After showing some examples of surface generation, the results are evaluated and conclusions are drawn.

§2. Surface Generation from Boundary Edges

2.1 General theory of surface generation

The fundamental idea is that a surface is created so that the interpolated point of a surface is obtained using rational blending functions for positional vectors and tangential vectors. Thus, the surface is a transfinite surface in consideration of the boundary positions and cross-boundary derivatives on the given boundary edges. It is a parametric surface created in mapping from a definition domain to 3D space $\mathbb{R}^2 \to \mathbb{R}^3$. Let us call the definition domain Ω and the boundary of the domain Γ. Using points $\alpha \in \Omega, \beta \in \Gamma$, $\boldsymbol{B}(\beta)$ is a positional vector and $\boldsymbol{D}(\beta)$ is a cross boundary derivative of a given boundary edge. These vectors specify the boundary conditions and have to be given as follows. A surface is represented as in (2) using blending functions at a point P:

$$\boldsymbol{B}(\beta) = \lim_{\alpha \to \beta} \boldsymbol{S}(\alpha), \qquad \boldsymbol{D}(\beta) = \lim_{\alpha \to \beta} \partial \boldsymbol{S}(\alpha)/\partial n, \tag{1}$$

$$\boldsymbol{S}(\alpha) = \oint_{\Gamma} \Phi(\alpha)(\boldsymbol{B}(\beta) + |\alpha - \beta| \boldsymbol{D}(\beta))). \tag{2}$$

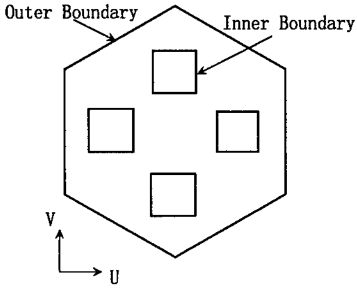

Fig. 1. Definition of the outer and inner boundaries.

2.2 Implementation method

Here the actual implementation method of N-sided surface-patch generation is described. Consider the normalized regular N-sided polygon in 2D space shown in Fig. 1 so that the foot length from an arbitrary point to each side is less than 1. This polygon is called "the outer boundary". Next, assume that one or more regular N-sided polygons are located within the outer boundary so that none of these polygons intersect with another. These polygons are called "inner boundaries". The closed domain D is defined as the area inside the outer boundary and outside the inner boundaries and is mapped to an N-sided surface patch in 3D space. Next prepare a pair of a boundary parameter and a distance parameter as follows.

(1) The distance parameter $d_{i,j}$ becomes 0 on side i, j, and varies from 0 to 1 according to the distance between point P and the side.

(2) The boundary parameter $b_{i,j}$ varies from 0 to 1 on side i, j of the given point P, and $b_{i,j}$ is given as the ratio of the adjacent distance parameters so that

$$b_{i,j} = d_{i,j-1}/(d_{i,j-1} + d_{i,j+1}). \tag{3}$$

Here i is the index of the outer boundary when $i = 0$, and the index of an inner boundary when $i \neq 0$. Here j is the index of a side of each boundary.

The values of u and v determine an arbitrary point $P(u, v)$ in the closed domain D. The pair of a distance parameter $d_{i,j}$ and a boundary parameter $b_{i,j}$ also determines the same point P. The blending function $\Phi_{l,m}$ for a side m of a boundary l is defined in the closed domain D as follows:

$$\Phi_{l,m}(u, v) = \frac{(1 - d_{l,m}^2)/d_{l,m}^2}{\sum_{p=0}^{H}\sum_{q=1}^{M_p}(1 - d_{p,q}^2)/d_{p,q}^2}. \tag{4}$$

Here the indices p and q relate to a side q of a boundary p. The boundary is the outer boundary when $p = 0$. Here H is the number of holes, and M_p is the number of sides of the boundary p. Also $\Phi_{l,m}$ is a function of u, v

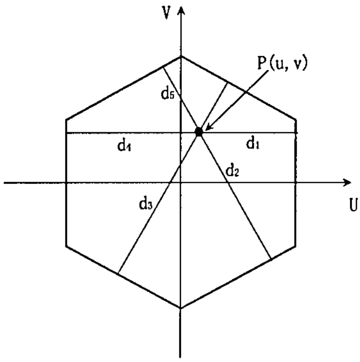

Fig. 2. Distance parameter for the outer boundary.

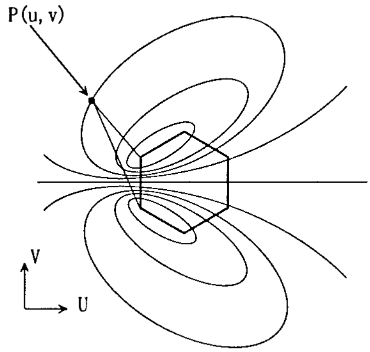

Fig. 3. Distance parameter and boundary parameter for an inner boundary.

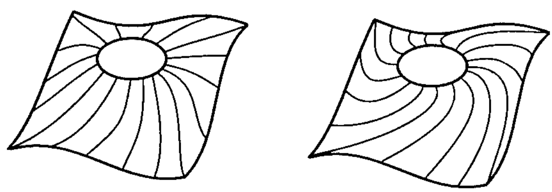

Fig. 4. Difference in surface generation according to definition domain.

and can also be a function of the pair of independent variables $b_{i,j}$ and $d_{i,j}$, because $d_{l,m}$ and $b_{l,m}$ are functions of $b_{i,j}$ and $d_{i,j}$. As shown in Fig. 2, the 3D boundary condition for a side j of a boundary i is given by a positional vector, $\boldsymbol{B}_{l,m}(b_{l,m})$ and a tangential vector, $\boldsymbol{D}_{l,m}(b_{l,m})$, where each vector is parameterized by the boundary parameter $b_{l,m}$. By giving the values of u and v, the variables $b_{p,q}$ and $d_{p,q}$ for side q of the boundary p are obtained, and $\boldsymbol{B}_{p,q}(b_{p,q})$ and $\boldsymbol{D}_{p,q}(b_{p,q})$ can be determined. The N-sided patch is given by the equation

$$\boldsymbol{S}(u,v) = \sum_{p=0}^{H}\sum_{q=1}^{M_p} \Phi_{p,q}(\boldsymbol{B}_{p,q}(b_{p,q}) + d_{p,q}\boldsymbol{D}_{p,q}(b_{p,q})). \tag{5}$$

§3. Problems with Conventional Surfaces

The previously proposed method removes the restrictions of a 4-sided patch [2]. This method enables a surface to be generated from the given boundary conditions (position, tangent vector), and is able to represent holes on the surface. However, it sometimes needs a manual transaction to generate a surface. For example, a generated surface sometimes becomes twisted or illegal when generating a surface with holes like the one shown in Fig. 4, although the

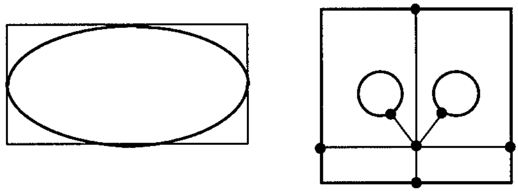

Fig. 5. Sabin's parametric domain.

surface satisfies the given boundary condition. Therefore, the earlier method must be revised in order to create a surface freely from arbitrary edges. It seems that the boundary in the 2D domain has to be similar to the 3D boundary in the 3D space. From such a point of view, Sabin proposed a surface patch using the definition space shown in Fig. 5. He also proposed a surface using dynamics from the same point of view with some restrictions from the same idea [5].

§4. Surface Generation from Boundary Edges by Reverse Mapping

In order to create a surface patch with concave edges and holes, the interpolation method has to be able to interpolate boundary edges on a plane at least without overlaps and protrusions. A transfinite surface blends sample points on the given boundary edges in an appropriate ratio. Therefore, the point to be interpolated exists in the convex hull of the sample points. Thus, it can be said that it is essentially difficult for a transfinite surface to generate a surface with a concave edge. Fig. 6a shows the boundary edges with a concave part on a plane in 3D space. It is desirable that point P is given as the point obtained by blending the sample points A1-A5 on each boundary edge. The point is in the convex hull of points A1-A5 because the blending functions have a value between 0 and 1.

Now consider the domain of Fig. 6b and Fig. 6c in UV-space. In case of Fig. 6b, the corresponding sample points b1-b5 are obtained for a point $p(u, v)$ and the sample points in 3D space become the points B1-B5 shown in Fig. 6d. It is thus possible that the blended point Q will be placed outside the boundary edges. In the case of Fig. 6c, sample points c1-c5 for point p will be obtained, and the corresponding points will be points C1-C5. Points C1 and C5 are affected strongly near the boundary, and point Q becomes an interpolated point and gives the good interpolated result shown in Fig. 6e. From the reasons outlined above, it can be said that the sample points are obtained near a point in the domain by using the definition boundary which resembles the given boundary edge. Since the sample points in 3D space for the point are given in appropriate ratios, a better interpolation can be realized compared with the conventional method in which a domain is a regular polygon. It can be also said that cross-boundary derivatives are helpful in avoiding a web in the concave part. The second term of (5) gives an effect of cross-tangential

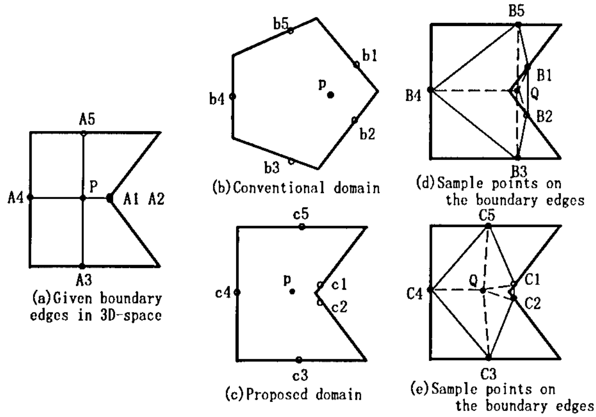

Fig. 6. Difference in sample points according to domain definition.

vectors to the surface. This term makes the surface point move according to the product of the distance parameter and the cross-boundary vector from the boundary.

4.1 Reverse mapping algorithm

In the previous section, it was stated that the boundary in a 2D definition space has to be similar to the shape of the 3D boundary edges. The following should be observed in constructing the domain boundary:

(1) The scale of 3D edges should reflect one of the domain edges.

(2) The topology of 3D edges should be the same as one of the domain edges.

(3) The angle of adjacent edges should reflect one of the domain edges.

Let us now consider a reverse mapping which satisfies these properties. Imagine a rubber surface spread over wires. By leaving it free, the wires will be straight. It would be ideal to use something like these wires as a 2D domain boundary. However this ideal mapping would be disadvantageous when considering the computational cost, so we selected a simple method of reverse mapping. Fig. 7 shows the algorithm. Each edge is connected at point P_k. Let the foot from the point to a plane be point Q_k. The plane can be obtained simply by solving the equation $\sum_{k=1}^{N}(P_k - P_k)^2$ so that it is minimized using the least square method. After obtaining the foot Q_k of the point to the plane, define the polygon which is constructed by Q_k as a 2D domain boundary.

4.2 Implementation method

In Section 2, the distance parameters and boundary parameters were defined in the normalized regular polygon of Fig. 2. The revised version of surface generation uses the same parameters, but uses non-normalized and irregular polygons with convex and concave parts. In order to define the distance

parameters for such polygons, we applied the previous calculation method used for the inner boundary to both the outer boundary and the inner boundaries. The boundary parameters are obtained by (3). Also the blending functions have to be revised because (4) is defined for the domain of a regular polygon. Next (7) is substituted for (5) in order to represent the concave figures shown in the next section:

$$\Phi_{l,m}(u,v) = \frac{1/d_{l,m}^2}{\sum_{p=0}^{H}\sum_{q=1}^{M_p} 1/d_{p,q}^2}, \tag{6}$$

$$\boldsymbol{S}(u,v) = \sum_{p=0}^{H}\sum_{q=1}^{M_p} \Phi_{p,q}(u,v)(\boldsymbol{B}_{p,q}(b_{p,q}) + d_{p,q}\boldsymbol{D}_{p,q}(b_{p,q}, d_{p,q})). \tag{7}$$

4.3 Theorems

Some characteristics of the blending function and the surface patch defined in the previous section are now discussed. The following theorems can be obtained from (6) and (7). Select a pair of independent variables $b_{i,j}$ and $d_{i,j}$.

Theorem 1. $\lim_{d_{i,j}\to 0} \Phi_{i,j}(u,v) = 1$, $\lim_{d_{l,m}\to 0} \Phi_{i,j}(u,v) = 0$ *when* $(l,m) \neq (i,j)$.

Theorem 2. $\dfrac{\partial \Phi_{l,m}(u,v)}{\partial b_{i,j}} = 0$ *and* $\dfrac{\partial \Phi_{l,m}(u,v)}{\partial d_{i,j}} = 0$.

Theorem 3. $\lim_{d_{i,j}\to 0} \boldsymbol{S}(u,v) = \boldsymbol{B}_{i,j}(b_{i,j})$, *and*

$$\lim_{d_{i,j}\to 0} \frac{\partial \boldsymbol{S}(u,v)}{\partial b_{i,j}} = \frac{\partial \boldsymbol{B}_{i,j}(b_{i,j})}{\partial b_{i,j}}, \quad \lim_{d_{i,j}\to 0} \frac{\partial \boldsymbol{S}(u,v)}{\partial d_{i,j}} = \lim_{d_{i,j}\to 0} \boldsymbol{D}_{i,j}(b_{i,j}, d_{i,j})$$

Proof:

$$\begin{aligned}
\lim_{d_{i,j}\to 0} \frac{\partial \boldsymbol{S}(u,v)}{\partial b_{i,j}} &= \lim_{d_{i,j}\to 0} \sum_{p=0}^{H}\sum_{q=1}^{M_p} [\frac{\partial \Phi_{p,q}}{\partial b_{i,j}}(\boldsymbol{B}_{p,q}(b_{p,q}) + d_{p,q}\boldsymbol{D}_{p,q}(b_{p,q}, d_{p,q})) + \\
&\quad \Phi_{p,q}(\frac{\partial \boldsymbol{B}(b_{p,q})}{\partial b_{i,j}} + \frac{\partial d_{p,q}}{\partial b_{i,j}}\boldsymbol{D}_{p,q}(b_{p,q}, d_{p,q}) + d_{p,q}\frac{\partial \boldsymbol{D}(b_{p,q}, d_{p,q})}{\partial b_{i,j}}] \\
&= \lim_{d_{i,j}\to 0} \Phi_{i,j}(\frac{\partial \boldsymbol{B}(b_{i,j})}{\partial b_{i,j}} + \frac{\partial d_{i,j}}{\partial b_{i,j}}\boldsymbol{D}_{i,j}(b_{i,j}, d_{i,j}) + d_{i,j}\frac{\partial \boldsymbol{D}(b_{i,j}, d_{i,j})}{\partial b_{i,j}}) = \frac{\partial \boldsymbol{B}(b_{i,j})}{\partial b_{i,j}} \\
\lim_{d_{i,j}\to 0} \frac{\partial \boldsymbol{S}(u,v)}{\partial d_{i,j}} &= \lim_{d_{i,j}\to 0} \sum_{p=0}^{H}\sum_{q=1}^{M_p} [\frac{\partial \Phi_{p,q}}{\partial d_{i,j}}(\boldsymbol{B}_{p,q}(b_{p,q}) + d_{p,q}\boldsymbol{D}_{p,q}(b_{p,q}, d_{p,q})) + \\
&\quad \Phi_{p,q}(\frac{\partial \boldsymbol{B}(b_{p,q})}{\partial d_{i,j}} + \frac{\partial d_{p,q}}{\partial d_{i,j}}\boldsymbol{D}_{p,q}(b_{p,q}, d_{p,q}) + d_{p,q}\frac{\partial \boldsymbol{D}(b_{p,q}, d_{p,q})}{\partial d_{i,j}}] \\
&= \lim_{d_{i,j}\to 0} \Phi_{i,j}(\frac{\partial \boldsymbol{B}(b_{i,j})}{\partial d_{i,j}} + \boldsymbol{D}_{i,j}(b_{i,j}, d_{i,j}) + d_{i,j}\frac{\partial \boldsymbol{D}(b_{i,j}, d_{i,j})}{\partial d_{i,j}}) = \boldsymbol{D}_{i,j}(b_{i,j}, d_{i,j}).
\end{aligned}$$

□

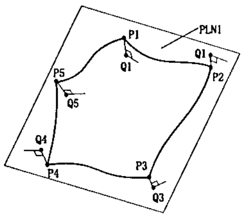

Fig. 7. Inverse mapping from 3D space into 2D domain.

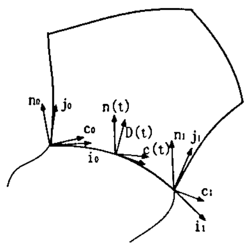

Fig. 8. Cross-boundary derivative.

§5. Boundary Condition and Surface Connection

5.1 Setting boundary conditions

A Coons patch must satisfy a compatibility condition. A Coons patch, which is constructed using a two surface patch in principle, does not guarantee the boundary condition without a compatibility condition. Gregory used a rational blending method, and invented a method of setting cross boundary derivatives freely so as to remove the inconvenience. A Gregory patch needs a compatibility condition for a positional boundary, and the twist vectors are discontinuous at the corners. In Little's patch the boundary conditions can be freely set for both position and tangent vector. The proposed surface does not cause the problems of the so-called compatibility condition. However, it is desirable that the tangent vector and the twist vector are continuous at the corners. Here, the method of setting cross-boundary derivatives at the corners is introduced. As shown in Fig. 8, two normal unit vectors $\boldsymbol{n}_0, \boldsymbol{n}_1$ are first calculated from the boundary derivatives $\boldsymbol{i}_0, \boldsymbol{j}_0, \boldsymbol{i}_1$, and $\boldsymbol{j}_1$. Here, the normal vector at a concave corner should be reversed. Define the two vectors $\boldsymbol{c}_0, \boldsymbol{c}_1$ obtained at the tips of the edge as $\boldsymbol{c}_0 = \boldsymbol{j}_0 \times \boldsymbol{n}_0, \boldsymbol{c}_1 = \boldsymbol{j}_1 \times \boldsymbol{n}_1$. The vectors $\boldsymbol{n}(t), \boldsymbol{c}(t)$ are from interpolating $\boldsymbol{n}_0, \boldsymbol{n}_1$ and $\boldsymbol{c}_0, \boldsymbol{c}_1$ respectively. Thus the cross boundary derivatives $\boldsymbol{D}(t)$ are given as $\boldsymbol{D}(t) = \boldsymbol{n}(t) \times \boldsymbol{c}(t)$.

In order to satisfy the compatibility condition for a tangent vector in a corner, the magnitude of a cross boundary vector has to been properly given. Since the boundary parameter $b_{i,j}$ for the edge(i,j) is defined as being between 0 and 1, the following reference has to be applied:

$$\left|\frac{\partial b_{i,j-1}}{\partial d_{i,j}}\right|_{\substack{d_{i,j}=0\\ b_{i,j}=0}} = \left.\frac{1}{d_{i,j-2}}\right|_{\substack{d_{i,j}=0\\ b_{i,j}=0}}, \qquad \left|\frac{\partial b_{i,j+1}}{\partial d_{i,j}}\right|_{\substack{d_{i,j}=0\\ b_{i,j}=1}} = \left.\frac{1}{d_{i,j+2}}\right|_{\substack{d_{i,j}=0\\ b_{i,j}=1}} \tag{8}$$

5.2 Boundary condition at concave corners and holes

A special transaction must be done in the cases of concave corners and holes so that the tangent vectors coincide with each other. As shown in Fig. 9, an edge is connected to the adjacent edges at concave and concave corners. Also an edge is connected to the adjacent edges at convex and concave corners in Fig. 10. For these cases, the cross boundary derivative has to be given as in

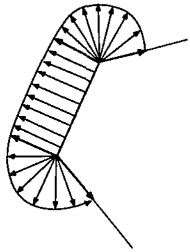

Fig. 9. Cross-boundary derivative in the concave-concave case.

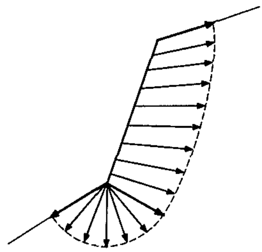

Fig. 10. Cross-boundary derivative in the convex-concave case.

these figures. Therefore, the derivative is given as a rational function. Giving the coefficients of (8) as $k0$ and $k1$ respectively, the tip vectors $\boldsymbol{c}_0, \boldsymbol{c}_1$ are given by (9). $\boldsymbol{c}_{00}$ and $\boldsymbol{c}_{11}$ are auxiliary vectors given to form the cross boundary vector. This is the reason why (5) is replaced with (7). For a surface patch with an isolated edge, the cross-boundary derivative can be set freely along the edge, because there is no adjacent surface around the isolated edge. Such a boundary edge is intended partially to trim a base patch:

$$\boldsymbol{c}_0 \equiv \frac{b\boldsymbol{c}_{00} + dk_0\boldsymbol{c}_0}{b + dk_0}, \qquad \boldsymbol{c}_1 \equiv \frac{(1-b)\boldsymbol{c}_{11} + dk_1\boldsymbol{c}_1}{(1-b) + dk_1} \tag{9}$$

§6. Example of Surface Generation

Figures 11 to 14 show examples of surface generation. Reverse mapping algorithms are applied to all of the surface generation. Compared with the old algorithm, it is unnecessary to modify the inner boundaries manually in a domain space. Fig. 11 shows an example with multiple holes. Fig. 12 shows an example with a ridge. The surface in Fig. 13 differs from the one in Fig. 14 in the shape of the hole, but both surfaces are generated in a desirable way.

§7. Conclusions

In this paper, a method of generating an N-sided patch with holes has been reviewed, and a revised method has been introduced. The following conclusions were obtained:

(1) The previous method has problems in generating a surface patch from boundary edges with concave parts and holes, because a transfinite surface essentially interpolates the sample points of given boundary edges.

(2) However, by using a reverse mapping from 3D space to 2D space, the shape of boundary edges becomes similar to the one in the 2D-space. This method relieves the above problems, and a surface can span arbitrary edges with holes and concave parts.

(3) In addition, this method can also represent isolated edges like ridges and valleys.

Acknowledgment. The author would like to thank Dr. M. A. Sabin of DAMPT, the University of Cambridge for his useful suggestions.

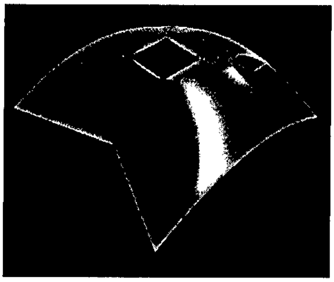

Fig. 11. Surface generation with two holes and a concave edge.

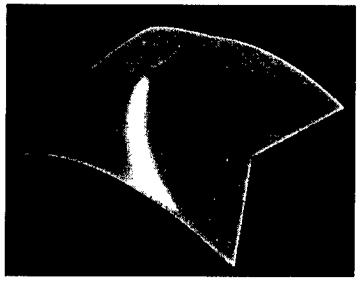

Fig. 12. Surface generation with a ridge.

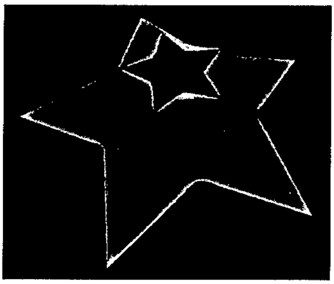

Fig. 13. Surface generation with concave parts and a hole.

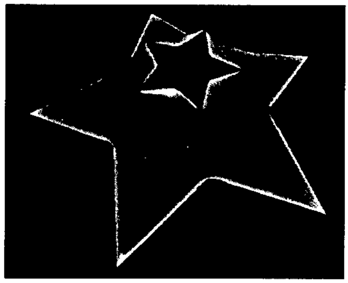

Fig. 14. The surface after the hole is rotated.

References

1. P. Malraison, A bibliography for N-sided surfaces, in Mathematics of Surfaces VIII Information Geometers, Winchester, 1998, 419–430.
2. K. Kato, Generation of N-sided surface patches with holes, Computer Aided Design **23** (1991), 676-683.
3. M. A. Sabin, Transfinite surface interpolation, in *The Mathematics of Surfaces VI*, G. Mullineux (ed.), Oxford, 1996, 517–537.
4. M. A. Sabin, Further transfinite surface developments, in Mathematics of Surfaces VIII, edited by Robert Cripps (ed.), Information Geometers, Winchester, 1998, 161-173.
5. M. A. Sabin, Two-sided patches suitable for inclusion in a B-Spline surface, in *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Lyche, and L. L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1998, 409–416.

Mitsubishi Electric Corporation, Industrial Electronics and Systems Lab
8-1-1, Tsukaguchi-Honmachi, Amagasaki, Hyogo, Japan
`kato@sdl.melco.co.jp`